Faster Private Release of Marginals on Small Databases[*]

Karthekeyan Chandrasekaran Justin Thaler Jonathan Ullman Andrew Wan

April 16, 2013

Abstract

We study the problem of answering *k-way marginal* queries on a database $D \in (\{0,1\}^d)^n$, while preserving differential privacy. The answer to a k-way marginal query is the fraction of the database's records $x \in \{0,1\}^d$ with a given value in each of a given set of up to k columns. Marginal queries enable a rich class of statistical analyses on a dataset, and designing efficient algorithms for privately answering marginal queries has been identified as an important open problem in private data analysis.

For any k, we give a differentially private online algorithm that runs in time

$$\min\left\{\exp\left(d^{1-\Omega(1/\sqrt{k})}\right), \exp\left(d/\log^{.99} d\right)\right\}$$

per query and answers any (possibly superpolynomially long and adaptively chosen) sequence of k-way marginal queries up to error at most ±.01 on every query, provided $n \gtrsim d^{.51}$. To the best of our knowledge, this is the first algorithm capable of privately answering marginal queries with a non-trivial worst-case accuracy guarantee on a database of size $\mathrm{poly}(d,k)$ in time $\exp(o(d))$.

Our algorithms are a variant of the private multiplicative weights algorithm (Hardt and Rothblum, FOCS '10), but using a different low-weight representation of the database. We derive our low-weight representation using approximations to the OR function by low-degree polynomials with coefficients of bounded L_1-norm. We also prove a strong limitation on our approach that is of independent approximation-theoretic interest. Specifically, we show that for any $k = o(\log d)$, any polynomial with coefficients of L_1-norm $\mathrm{poly}(d)$ that pointwise approximates the d-variate OR function on all inputs of Hamming weight at most k must have degree $d^{1-O(1/\sqrt{k})}$.

[*] Harvard University, School of Engineering and Applied Sciences. Email: {karthe, jthaler, jullman, atw12}@seas.harvard.edu. Karthekeyan Chandrasekaran is supported by Simons Fellowship. Justin Thaler is supported by an NSF Graduate Research Fellowship and NSF grants CNS-1011840 and CCF-0915922. Jonathan Ullman is supported by NSF grant CNS-1237235 and a Siebel Scholarship. Andrew Wan is supported by NSF grant CCF-0964401 and NSFC grant 61250110218.

1 Introduction

Consider a database $D \in (\{0,1\}^d)^n$ in which each of the $n(= |D|)$ rows corresponds to an individual's record, and each record consists of d binary attributes. The goal of privacy-preserving data analysis is to enable rich statistical analyses on the database while protecting the privacy of the individuals. In this work, we seek to achieve *differential privacy* [DMNS06], which guarantees that no individual's data has a significant influence on the information released about the database.

One of the most important classes of statistics on a dataset is its *marginals*. A marginal query is specified by a set $S \subseteq [d]$ and a pattern $t \in \{0,1\}^{|S|}$. The query asks, "What fraction of the individual records in D has each of the attributes $j \in S$ set to t_j?" A major open problem in privacy-preserving data analysis is to *efficiently* create a differentially private summary of the database that enables analysts to answer each of the 3^d marginal queries. A natural subclass of marginals are *k-way marginals*, the subset of marginals specified by sets $S \subseteq [d]$ such that $|S| \leq k$.

Privately answering marginal queries is a special case of the more general problem of privately answering *counting queries* on the database, which are queries of the form, "What fraction of individual records in D satisfy some property q?" Early work in differential privacy [DN03, BDMN05, DMNS06] showed how to privately answer any set of counting queries $\mathcal{Q}$ approximately, and yet with good accuracy (say, within ±.01 of the true answer), by perturbing the answers with appropriately calibrated noise, provided $|D| \gtrsim |\mathcal{Q}|^{1/2}$.

However, in many settings data is difficult or expensive to obtain, and the requirement that $|D| \gtrsim |\mathcal{Q}|^{1/2}$ is too restrictive. For instance, if the query set $\mathcal{Q}$ includes all k-way marginal queries then $|\mathcal{Q}| \geq d^{\Theta(k)}$, and it may be impractical to collect enough data to ensure $|D| \gtrsim |\mathcal{Q}|^{1/2}$, even for moderate values of k. Fortunately, a remarkable line of work initiated by Blum et. al. [BLR08] and continuing with [DNR$^+$09, DRV10, RR10, HR10, HLM12, GRU12, JT12], has shown how to privately release approximate answers to any set of counting queries, even when $|\mathcal{Q}|$ is *exponentially larger* than $|D|$. For example, the *online private multiplicative weights algorithm* of Hardt and Rothblum [HR10] gives accurate answers to any (possibly adaptively chosen) sequence of queries $\mathcal{Q}$ provided $|D| \gtrsim \sqrt{d} \log |\mathcal{Q}|$. Hence, if the sequence consists of all k-way marginal queries, then the algorithm will give accurate answers provided $|D| \gtrsim k\sqrt{d}$. Unfortunately, all of these algorithms have running time at least 2^d per query, even in the simplest setting where $\mathcal{Q}$ is the set of 2-way marginals.

Given this state of affairs, it is natural to seek *efficient* algorithms capable of privately releasing approximate answers to marginal queries even when $|D| \ll d^k$. The most efficient algorithm known for this problem answers *all* k-way marginal queries in time $d^{O(\sqrt{k})}$ and answers every conjunction query to within ±.01 provided $|D| \gtrsim d^{O(\sqrt{k})}$ [TUV12].[1]

Even though $|D|$ can be much smaller than $|\mathcal{Q}|^{1/2}$, a major drawback of this algorithm, and other efficient algorithms for releasing marginals (e.g. [GHRU11, CKKL12, HRS12, FK13]) is that the database still must be significantly larger than $\tilde{\Theta}(k\sqrt{d})$, which we know would suffice for inefficient algorithms. Recent experimental work of Hardt et al. [HLM12] demonstrates that for some databases of interest, even the 2^d-time private multiplicative weights algorithm is practical, and also shows that more efficient algorithms based on adding independent noise do not provide good accuracy for these databases. Motivated by these findings, we believe that an important approach to designing practical algorithms is to achieve a minimum database size comparable to that of private multiplicative weights, and seek to optimize the running time of the algorithm as much as possible. In this paper we give the first algorithms for privately answering marginal queries

[1] More precisely, the algorithm in [TUV12] runs in time $d^{O(\sqrt{k})}$ and releases a summary from which an analyst can compute the answer to any k-way marginal query in time $d^{O(\sqrt{k})}$.

Paper	Running Time per Query	Database Size
[DN03, BDMN05, DMNS06]	$O(1)$	$O(d^{k/2})$
[BLR08, DNR$^+$09, DRV10, HLM12]	$2^{O(d)}$	$\tilde{O}(k\sqrt{d})$
[TUV12]	$d^{O(\sqrt{k})}$	$d^{O(\sqrt{k})}$
This paper	$2^{O(d/\log^{.99} d)}$	$kd^{.5+o(1)}$
This paper	$2^{d^{1-\Omega(1/\sqrt{k})}}$	$kd^{.51}$

Table 1: Summary of prior results on differentially private release of k-way marginals with error $\pm.01$ on every marginal. Note that the running time ignores dependence on the database size, privacy parameters, and the time required to evaluate the query non-privately.

for this parameter regime.

1.1 Our Results

In this paper we give faster algorithms for privately answering marginal queries on databases of size $\tilde{O}(d^{.51}/\varepsilon)$, which is nearly the smallest a database can be while admitting any differentially private approximation to marginal queries.[2]

Theorem 1.1. *There exists a constant $C > 0$ such that for every $k, d, n \in \mathbb{N}$, $k \leq d$, and every $\varepsilon, \delta > 0$, there is an (ε, δ)-differentially private online algorithm that, on input a database $D \in (\{0,1\}^d)^n$, runs in time*

$$\min\left\{\exp\left(d^{1-1/C\sqrt{k}}\right), \exp\left(d/\log^{.99} d\right)\right\}$$

per query and answers any sequence $\mathcal{Q}$ of (possibly adaptively chosen) k-way marginal queries on D up to an additive error of at most $\pm.01$ on every query with probability at least 0.99, provided that $|D| = \Omega(d^{.51} \log|\mathcal{Q}| \log(1/\delta)/\varepsilon)$.

See Table 1 for a comparison of relevant results on privately answering marginal queries.

When k is much smaller than d, it may be useful to view our algorithm as an offline algorithm for releasing answers to all k-way marginal queries. This offline algorithm can be obtained simply by requesting answers to each of the $d^{\Theta(k)}$ distinct k-way marginal queries from the online mechanism. In this case we obtain the following corollary.

Corollary 1.2. *There exists a constant $C > 0$ such that for every $k, d, n \in \mathbb{N}$, $k = O(d/\log d)$, and every $\varepsilon, \delta > 0$, there is an (ε, δ)-differentially private online algorithm that, on input a database $D \in (\{0,1\}^d)^n$, runs in time*

$$\min\left\{\exp\left(d^{1-1/C\sqrt{k}}\right), \exp\left(d/\log^{.99} d\right)\right\}$$

per query and, with probability at least 0.99, releases answers to every k-way marginal query on D up to an additive error of at most $\pm.01$, provided that $|D| = \Omega(kd^{.51} \log(1/\delta)/\varepsilon)$.

We make a few additional remarks about our results:

Remark 1. When $k = \Omega(\log^2 d)$, the minimum database size requirement can be improved to $|D| \geq Ckd^{.5+o(1)} \log(1/\delta)/\varepsilon$ (for some universal constant $C > 0$), but we have stated the theorems with a looser bound for simplicity. Here the $o(1)$ means as a function of d.

[2] There is no differentially private algorithm that answers even 1-way marginal queries with non-trivial accuracy on a database of size $o(\sqrt{d}/\log d)$ [Ull12, Vad]

Remark 2. Our algorithm can be modified so that instead of releasing approximate answers to each k-way marginal explicitly, it releases a summary of the database of size $\tilde{O}(kd^{.01})$ from which an analyst can compute an approximate answer to any k-way marginal in time $\tilde{O}(kd^{1.01})$.

A key ingredient in our algorithm is a new approximate representation of the database by a low-degree polynomial of low weight. Our approximate database representation relies upon the construction of a low-degree polynomial of weight poly(d) that approximates the d-variate OR function on all inputs of Hamming weight at most k. Constructing such a polynomial of lower degree and similar weight would immediately yield a faster data release algorithm with a similar accuracy guarantee. Unfortunately, we prove a lower bound showing that no such polynomial exists. This lower bound may be of independent approximation-theoretic interest.

Theorem 1.3. *Let $OR_d : \{-1,1\}^d \to \{-1,1\}$ denote the OR function on d variables, and for any vector $x \in \{-1,1\}^d$, let $|x|$ denote $\sum_{i=1}^d \mathbf{1}_{\{x_i=-1\}}$, the number of coordinates of x equal to -1. Fix $k = o(\log d)$, and let p be a real d-variate polynomial satisfying $|p(x) - OR_d(x)| \leq 1/6$ for all $x \in \{-1,1\}^d$ with $|x| \leq k$. If the sum of the absolute values of the coefficients of p is bounded by $d^{O(1)}$, then the degree of p is at least $d^{1-O(1/\sqrt{k})}$.*

We note that our lower bound limits the applicability of our algorithm design technique when we use low-degree polynomials (i.e. linear combinations of low-degree monomials) to uniformly approximate all disjunctions over d variables (and in turn, to represent the database). Several natural candidates (e.g. the set of small-width conjunctions) can themselves be computed exactly by a low-weight polynomial of low-degree, and thus our lower bound applies to these feature spaces as well.

It is an interesting open question to decide whether or not there exists a smaller "feature-space" of functions other than low-degree parities or low-width conjunctions such that every disjunction over d variables can be uniformly approximated by a low-weight linear combination of features. An affirmative answer to this question would immediately yield a more efficient algorithm than ours with a similar accuracy guarantee.

1.2 Techniques

We now describe the algorithm promised by Theorem 1.1. For notational convenience, we focus on *monotone k-way disjunction queries.* However, our results extend straightforwardly to general non-monotone k-way marginal queries via simple transformations on the database and queries. A monotone k-way disjunction is specified by a set $S \subseteq [d]$ of size k and asks what fraction of records in D have at least one of the attributes in S set to 1.

Following the approach taken in many prior works on privately releasing conjunctions and other families of queries, we view the database as a function $f_D \colon \{-1,1\}^d \to [0,1]$, in which each input vector $s \in \{-1,1\}^d$ is interpreted as the indicator vector of a set $S \subseteq \{1,\ldots,d\}$ (with $s_i = -1$ iff $i \in S$), and $f_D(s)$ equals the evaluation of the conjunction query specified by S on the database D.

The starting point for our algorithm is the online private multiplicative weights algorithm (PMW) [HR10], which has running time 2^d per query and answers any sequence of *arbitrary counting queries* provided $|D| \gtrsim \sqrt{d}\log|\mathcal{Q}|$. Gupta, Roth, and Ullman [GRU12] introduced the "IDC framework" — capturing PMW and other algorithms — for designing differentially private online algorithms and, in particular, showed that such an algorithm can be derived from any *online learning algorithm* that may not necessarily be privacy preserving.

Informally, an online learning algorithm is one that takes a (possibly adaptively chosen) sequence of inputs $s_1, s_2, \ldots$ and returns answers $a_1, a_2, \ldots$ to each, representing "guesses" about the values

$f_D(s_1), f_D(s_2), \ldots$ for the unknown function f_D. After making each guess a_i, the learner is given some information about the value of $f_D(s_i)$. The quantities of interest are the running time required by the online learner to produce each guess a_i and the number of "mistakes" made by the learner, which is roughly the number of periods i in which a_i is far from $f_D(s_i)$. Ultimately, for the differentially private algorithm derived in the IDC framework, the notion of far will correspond to the error, the per query running time will essentially be equal to the running time of the online learning algorithm, and the minimum database size required by the private algorithm will be proportional to the number of mistakes.

The algorithm of Hardt and Rothblum [HR10], is based on an online learning algorithm that runs in time 2^d and makes $O(d)$ mistakes. A standard approach to obtaining a faster online learning algorithm that still makes few mistakes is to use a *polynomial approximation* to the target function f_D. Indeed, it is well-known that if f_D can be approximated to high accuracy in the L_∞ norm by a d-variate polynomial $p_D\colon \{-1,1\}^d \to \mathbb{R}$ of degree t and L_1-weight W (defined to be the sum of the absolute values of the coefficients), then there is an online learning algorithm that runs in time $\mathrm{poly}\left(\binom{d}{t}\right)$ and makes $O(Wd)$ mistakes. Thus, if $t \ll d$, the running time of such an online learning algorithm will be significantly less than 2^d and the number of mistakes (and thus the minimum database size of the resulting private algorithm) will only blow up by a factor of W.

Our goal is therefore to demonstrate that for any database D, there is a low-degree, low-weight polynomial p_D such that $|p_D(s) - f_D(s)|$ is small for all vectors $s \in \{-1,1\}^d$ corresponding to monotone k-way disjunction queries. To accomplish this, it is sufficient to construct a low-degree, low-weight polynomial that can approximate the d-variate OR function on inputs of Hamming weight at most k (those that have -1 in at most k indices). We achieve this, showing that for any k there is a suitable polynomial of degree $d^{1-\Omega(1/\sqrt{k})}$ and weight $d^{.01}$. For larger values of k—for which the previous bound becomes trivial—we show that a suitable polynomial exists with degree $d/\log^{.99} d$ and weight $d^{o(1)}$.

We also prove a new approximation-theoretic lower bound of independent interest. The lower bound suggests that we may have already reached the limit of our approach for designing efficient private data release algorithms. Specifically, we show that for any $k = o(\log d)$, any polynomial p of weight $\mathrm{poly}(d)$ that satisfies $|p(s) - \mathrm{OR}(s)| \leq 1/6$ for all inputs $s \in \{-1,1\}^d$ of Hamming weight at most k must have degree $d^{1-O(1/\sqrt{k})}$. We prove our lower bound by expressing the problem of constructing such a low-weight, low-degree polynomial p as a linear program, and exhibiting an explicit solution to the dual of this linear program. Our proof is inspired by recent work (cf. Sherstov [She09, She11, She12b] and Bun and Thaler [BT13]) that proves new approximation-theoretic lower bounds via the construction of dual solutions to appropriate linear programs.

Other Results on Privately Releasing Marginals While we have focused on accurately answering *every* k-way marginal query, or more generally every query in a sequence of marginal queries, several other works have considered more relaxed notions of accuracy. These works show how to efficiently release a summary of the database from which an analyst can efficiently compute an approximate answer to marginal queries, with the guarantee that the *average* error of a marginal query is at most .01, when the query is chosen from a particular distribution. In particular, Feldman and Kothari [FK13] achieve small average error over the uniform distribution with running time and database size $\tilde{O}(d^2)$; Gupta et al. [GHRU11] achieve small average error over any product distribution with running time and minimum database size $\mathrm{poly}(d)$; finally Hardt et al. [HRS12] show how to achieve small average error over arbitrary distributions with running time and minimum database size $2^{\tilde{O}(d^{1/3})}$. All of these results are based on the approach of learning the function f_D.

Several works have also considered information theoretic bounds on the minimum database size required to answer k-way marginals. Kasiviswanathan et al. [KRSU10] showed that the minumum database size must be at least $\min\{\frac{1}{\alpha^2}, \frac{d^{k/2}}{\alpha}\}$ to answer all k-way marginals with error $\pm\alpha$. In the regime we consider where $\alpha = \Omega(1)$, their results do not give a non-trivial lower bound.

Relationship with Hardness Results for Differential Privacy. Ullman [Ull12] (building on the results of Dwork et al. [DNR+09]), showed that any $2^{o(d)}$-time differentially private algorithm that answers *arbitrary* counting queries can only give accurate answers if $|D| \gtrsim |\mathcal{Q}|^{1/2}$, assuming the existence of exponentially hard one-way functions. Our algorithms have running time $2^{o(d)}$ and are accurate when $|D| \ll |\mathcal{Q}|^{1/2}$, and thus show a separation between answering marginal queries and answering arbitrary counting queries.

When viewed as an offline algorithm for answering all k-way marginals, our algorithm will return a list of values containing answers to each k-way marginal query. It would in some cases be more attractive if we could return a *synthetic database*, which is a new database $\widehat{D} \in (\{0,1\}^d)^{\widehat{n}}$ whose rows are "fake", but such that $\widehat{D}$ approximately preserves many of the statistical properties of the database D (e.g. all the marginals). Some of the previous work on counting query release has provided synthetic data, starting with Barak et. al. [BCD+07] and including [BLR08, DNR+09, DRV10, HLM12].

Unfortunately, Ullman and Vadhan [UV11] (building on [DNR+09]) have shown that no differentially private sanitizer with running time poly(d) can take a database $D \in (\{0,1\}^d)^n$ and output a private synthetic database $\widehat{D}$, all of whose 2-way marginals are approximately equal to those of D (assuming the existence of one-way functions). They also showed that there is no differentially private sanitizer with running time $2^{d^{1-\Omega(1)}}$ can output a private synthetic database, all of whose 2-way marginals are approximately equal to those of D. Our algorithms indeed achieve this running time and accuracy guarantee when releasing k-way marginals for constant k, and thus it is inherent that our algorithms do not generate synthetic data.

Relationship with Results in Learning and Approximation Theory. Servedio, Tan, and Thaler [STT12] focused on developing low-weight, low-degree polynomial threshold functions (PTFs) for decision lists, motivated by applications in computational learning theory. As an intermediate step in their PTF constructions, they constructed low-weight, low-degree polynomials that approximate the OR function on *all* Boolean inputs. Our construction of lower-weight, lower-degree polynomials that approximate the OR function on low Hamming weight inputs is inspired by and builds on Servedio et al.'s construction for the entire Boolean hypercube.

The proof of our lower bound is inspired by recent work that has established new approximate degree lower bounds via the construction of dual solutions to certain linear programs. In particular, Sherstov [She09] showed that approximate degree and PTF degree behave roughly multiplicatively under function composition, while Bun and Thaler [BT13] gave a refinement of Sherstov's method in order to resolve the approximate degree of the two-level AND-OR tree, and also gave an explicit dual witness for the approximate degree of any symmetric Boolean function. We extend these lower bounds along two directions: (1) we show degree lower bounds that take into account the size of the coefficients of the approximating polynomial, and (2) our lower bounds hold even when we only require the approximation to be accurate on inputs of low Hamming weight, while prior work only considered approximations that are accurate on the entire Boolean hypercube.

Some prior work has studied the degree of polynomials that point-wise approximate *partial* Boolean functions [She12b, She12a]. Here, a function $f : Y \to \mathbb{R}$ is said to be partial if its domain Y is a strict subset of $\{-1,1\}^d$, and a polynomial p is said to ϵ-approximate f if

1. $|f(x) - p(x)| \le \epsilon$ for all $x \in Y$, and
2. $|p(x)| \le 1 + \epsilon$ for all $x \in \{-1,1\}^d \setminus Y$.

In contrast, our lower bounds apply even in the absence of Condition 2, i.e. when $p(x)$ is allowed to take arbitrary values in $\{-1,1\}^d \setminus Y$.

Finally, while our motivation is private data release, our approximation theoretic results are similar in spirit to recent work of Long and Servedio [LS13], who are motivated by applications in computational learning theory. Long and Servedio consider halfspaces h defined on inputs of small Hamming weight, and (using different techniques very different from ours) give upper and lower bounds on the weight of these halfspaces when represented as linear threshold functions.

Organization. In Section 3, we describe our differentially private online algorithm and show that it yields the claimed accuracy given the existence of sufficiently low-weight polynomials that uniformly approximate the d-variate OR function on inputs of low Hamming weight. The results of this section are a combination of known techniques in differential privacy [RR10, HR10, GRU12] and learning theory (see e.g. [KS04]). Readers familiar with these literatures may prefer to skip Section 3 on first reading. In Section 4, we give our polynomial approximations to the OR function, both on low-weight inputs and on inputs from the entire Boolean cube. Finally, in Section 5, we state and prove our lower bounds for polynomial approximations to the OR function on restricted inputs.

2 Preliminaries

2.1 Differentially Private Sanitizers

Let a *database* $D \in \mathcal{X}^n$ be a collection of n rows $x^{(1)}, \dots, x^{(n)}$ from a *data universe* $\mathcal{X}$. We say that two databases $D, D' \in \mathcal{X}^n$ are *adjacent* if they differ only on a single row, and we denote this by $D \sim D'$.

Let $\mathcal{A} : \mathcal{X}^n \to \mathcal{R}$ be an algorithm that takes a database as input and outputs some data structure in $\mathcal{R}$. We are interested in algorithms that satisfy *differential privacy.*

Definition 2.1 (Differential Privacy [DMNS06])**.** An algorithm $\mathcal{A}\colon \mathcal{X}^n \to \mathcal{R}$ is (ε,δ)-*differentially private* if for every two adjacent databases $D \sim D' \in \mathcal{X}^n$ and every subset $S \subseteq \mathcal{R}$,

$$\Pr\left[\mathcal{A}(D) \in S\right] \le e^{\varepsilon} \Pr\left[\mathcal{A}(D') \in S\right] + \delta.$$

Since a sanitizer that always outputs $\perp$ satisfies Definition 2.1, we focus on sanitizers that are accurate. In particular, we are interested in sanitizers that give accurate answers to *counting queries.* A counting query is defined by a boolean predicate $q\colon \mathcal{X} \to \{0,1\}$. Abusing notation, we define the evaluation of the query q on a database $D \in \mathcal{X}^n$ to be $q(D) = \frac{1}{n}\sum_{i=1}^{n} q(x^{(i)})$. Note that the value of a counting query is in $[0,1]$. We use $\mathcal{Q}$ to denote a set of counting queries.

For the purposes of this work, we assume that the range of $\mathcal{A}$ is simply $\mathbb{R}^{|\mathcal{Q}|}$. That is, $\mathcal{A}$ outputs a list of real numbers representing answers to each of the specified queries.

Definition 2.2 (Accuracy)**.** The output of $\mathcal{A}(D)$, $a = (a_q)_{q \in \mathcal{Q}}$, is α-*accurate* for the query set $\mathcal{Q}$ if

$$\forall q \in \mathcal{Q},\ |a_q - q(D)| \le \alpha$$

A sanitizer is (α,β)-*accurate* for the query set $\mathcal{Q}$ if for every database D, $\mathcal{A}(D)$ outputs a such that with probability at least $1-\beta$, a is α-accurate for $\mathcal{Q}$, where the probability is taken over the coins of $\mathcal{A}$.

We remark that the definition of both differential privacy and (α, β)-accuracy extend straightforwardly to the online setting. Here the algorithm receives a sequence of ℓ (possibly adaptively chosen) queries from $\mathcal{Q}$ and must give an answer to each before seeing the rest of the sequence. Here we require that with probability at least $1 - \beta$, *every* answer given is within $\pm\alpha$ of the true answer on D. See e.g. [HR10] for a full treatment of the online setting.

2.2 Query Function Families

Given a set of queries of interest, $\mathcal{Q}$ (e.g. all marginal queries), we think of the database D as specifying a function f_D mapping queries q to their answers $q(D)$, which we call the *$\mathcal{Q}$-representation of D*. We now describe this transformation more formally:

Definition 2.3 ($\mathcal{Q}$-Function Family)**.** Let $\mathcal{Q} = \{q_y\}_{y \in Y_{\mathcal{Q}} \subseteq \{-1,1\}^m}$ be a set of counting queries on a data universe $\mathcal{X}$, where each query is indexed by an m-bit string. We define the *index set of $\mathcal{Q}$* to be the set $Y_{\mathcal{Q}} = \{y \in \{-1,1\}^m \mid q_y \in \mathcal{Q}\}$.

We define the *$\mathcal{Q}$-function family* $\mathcal{F}_{\mathcal{Q}} = \{f_x : \{-1,1\}^m \to [0,1]\}_{x \in \mathcal{X}}$ as follows: For every possible database row $x \in \mathcal{X}$, the function $f_{\mathcal{Q},x} : \{-1,1\}^m \to [0,1]$ is defined as $f_{\mathcal{Q},x}(y) = q_y(x)$. Given a database $D \in \mathcal{X}^n$ we define the function $f_{\mathcal{Q},D} : \{-1,1\}^m \to [0,1]$ where $f_{\mathcal{Q},D}(q) = \frac{1}{n}\sum_{i=1}^n f_{\mathcal{Q},x^{(i)}}(q)$. When $\mathcal{Q}$ is clear from context we will drop the subscript $\mathcal{Q}$ and simply write f_x, f_D, and $\mathcal{F}$.

When $\mathcal{Q}$ is the set of all monotone k-way disjunctions on a database $D \in (\{0,1\}^d)^n$, the queries are defined by sets $S \subseteq [d]$, $|S| \le k$. In this case, we represent each query by the d-bit $-1/1$ indicator vector y_S of the set S, where $y_S(i) = -1$ if and only if $i \in S$. Thus, y_S has at most k entries that are -1. Hence, we can take $m = d$ and $Y_{\mathcal{Q}} = \left\{ y \in \{-1,1\}^d \mid \sum_{j=1}^d \mathbf{1}_{\{y_i=-1\}} \le k \right\}$.

2.3 Low-Weight Polynomial Approximations

Given an m-variate real polynomial $p\colon \{-1,1\}^m \to \mathbb{R}$,

$$p(y) = \sum_{S \subseteq [m]} c_S \cdot \prod_{i \in S} y_i,$$

we define the degree, weight $w(.)$ and *non-constant weight* $w^*(.)$ of the polynomial as follows:

$$\begin{aligned} \deg(p) &:= \max\{|S| : S \subseteq [m], c_S \neq 0\}, \\ w(p) &:= \sum_{S \subseteq [m]} |c_S|, and \\ w^*(p) &:= \sum_{S \subseteq [m], S \neq \emptyset} |c_S|. \end{aligned}$$

We use $\binom{[m]}{\le t}$ to denote $\{S \subseteq [m] \mid |S| \le t\}$ and $\binom{m}{\le t} = \left|\binom{[m]}{\le t}\right| = \sum_{j=0}^t \binom{m}{j}$.

In many cases, the functions $f_{\mathcal{Q},x} : \{-1,1\}^m \to \{0,1\}$ can be approximated well on all the indices in $Y_{\mathcal{Q}}$ by a family of polynomials with low degree and low weight. Formally and more generally:

Definition 2.4 (Restricted Approximation by Polynomials)**.** Given a function $f\colon Y \to \mathbb{R}$, where $Y \subseteq \mathbb{R}^m$, and a subset $Y' \subseteq Y$, we denote the restriction of f to Y' by $f|_{Y'}$. Given an m-variate real polynomial p, we say that p is a *γ-approximation* to the restriction $f|_{Y'}$, if $|f(y) - p(y)| \le \gamma$ $\forall y \in Y'$. Notice there is no restriction whatsoever placed on $p(y)$ for $y \in Y \setminus Y'$.

Given a family of m-variate functions $\mathcal{F} = \{f_x \colon Y \to \mathbb{R}\}_{x \in \mathcal{X}}$, where $Y \subseteq \mathbb{R}^m$, a set $Y' \subseteq Y$ and a family $\mathcal{P}$ of m-variate real polynomials, we say that the family $\mathcal{P}$ is a *γ-approximation to $\mathcal{F}|_{Y'}$* if for every $x \in \mathcal{X}$, there exists $p_x \in \mathcal{P}$ that is a γ-approximation to $f_x|_{Y'}$.

Let $H_{m,k} = \{x \in \{-1,1\}^m : \sum_{i=1}^m (1 - x_i)/2 \le k\}$ denote the set of inputs of Hamming weight at most k. We view the d variate OR function, OR_d as mapping inputs from $\{-1,1\}^d$ to $\{-1,1\}$, with the convention that -1 is TRUE and 1 is FALSE. Let $\mathcal{P}_{t,W}$ denote the family of all m-variate real polynomials of degree t and weight W. For the upper bound, we will show that for certain small values of t and W, the family $\mathcal{P}_{t,W}$ is a γ-approximation to the family of all disjunctions restricted to $H_{m,k}$.

Fact 2.5. *If $\mathcal{Q}$ is the set of all monotone k-way disjunctions on a database $D \in (\{0,1\}^d)^n$, $\mathcal{F}$ is its function family, and $Y = H_{d,k}$ is its index set, then $\mathcal{P}_{t,W}$ is a γ-approximation to the restriction $\mathcal{F}|_Y$ if and only if there is a degree t polynomial of weight W that γ-approximates $OR_d|_{H_{d,k}}$.*

The fact follows easily by observing that for any $x \in \{0,1\}^d$, $y \in \{-1,1\}^d$,

$$f_x(y) = \bigvee_{i \in x} \mathbf{1}_{\{y_i = -1\}} = \mathrm{OR}_d(y_1^{x_1}, \dots, y_d^{x_d}).$$

For the lower bound, we will show that any collection of polynomials with small weight that is a γ-approximation to the family of disjunctions restricted to $H_{m,k}$ should have large degree. We need the following definitions:

Definition 2.6 (Approximate Degree)**.** Given a function $f \colon Y \to \mathbb{R}$, where $Y \subseteq \mathbb{R}^m$, the γ-approximate degree of f is

$$\deg_\gamma(f) := \min\{d : \exists \text{ real polynomial } p \text{ that is a } \gamma\text{-approximation to } f,\ \deg(p) = d\}.$$

Analogously, the (γ, W)-approximate degree of f is

$$\deg_{(\gamma,W)}(f) := \min\{d : \exists \text{ real polynomial } p \text{ that is a } \gamma\text{-approximation to } f,\ \deg(p) = d,\ w(p) \le W\}.$$

It is clear that $\deg_\gamma(f) = \deg_{(\gamma,\infty)}(f)$.

We let $w^*(f,t)$ denote the *degree-t non-constant margin weight* of f, defined to be:

$$w^*(f,t) := \min\{w^*(p) : \deg(p) \le t, f(y)p(y) \ge 1\ \forall\ y \in Y\}.$$

The above definitions extend naturally to the restricted function $f|_{Y'}$.

Our definition of non-constant margin weight is closely related to the well-studied notion of the degree-t polynomial threshold function (PTF) weight of f (see e.g. [She11]), which is defined as $\min_p w(p)$, where the minimum is taken over all degree-t polynomials p with integer coefficients, such that $f(x) = \mathrm{sign}(p(x))$ for all $x \in \{-1,1\}^d$. Often, when studying PTF weight, the requirement that p have integer coefficients is used only to ensure that p has non-trivial margin, i.e. that $|p(x)| \ge 1$ for all $x \in \{-1,1\}^d$; this is precisely the requirement captured in our definition of non-constant margin weight. We choose to work with margin weight because it is a cleaner quantity to analyze using linear programming duality; PTF weight can also be studied using LP duality, but the integrality constraints on the coefficients of p introduces an integrality gap that causes some loss in the analysis (see e.g. Sherstov [She11, Theorem 3.4] and Klauck [Kla11, Section 4.3]).

3 From Low-Weight Approximations to Private Data Release

In this section we show that low-weight polynomial approximations imply data release algorithms that provide approximate answers even on small databases. Informally, if the family of low-weight, low-degree m-variate polynomials $\mathcal{P}_{t,W}$ $(1/400)$-approximates $\mathcal{F}_{\mathcal{Q}}$, then there is a differentially private online algorithm algorithm with running time $\mathrm{poly}\big(\binom{m}{\leq t}, |\mathcal{Q}|\big)$ that releases answers to every query sequence of ℓ queries in $\mathcal{Q}$ within error $\pm.01$ as long as $n \gtrsim W\sqrt{m}\log \ell/\varepsilon$.

The results in this section can be assembled from known techniques in the design and analysis of differentially private algorithms and online learning algorithms. We include them for completeness, as to our knowledge they do not explicitly appear in the privacy literature.

We construct and analyze the algorithm in two steps. First, we use standard arguments to show that the non-private multiplicative weights algorithm can be used to construct a suitable online learning algorithm for $f_{\mathcal{Q},D}$ whenever $f_{\mathcal{Q},D}$ can be approximated by a low-weight, low-degree polynomial. Here, a suitable online learning algorithm is one that fits into the IDC framework of Gupta et al. [GRU12]. We then apply the generic conversion from IDCs to differentially private online algorithms [RR10, HR10, GRU12] to obtain our algorithm.

3.1 IDCs

We start by providing the relevant background on the *iterative database construction* framework. To cut down on notational clutter, we give the definition specialized to the case where the database is to be represented by a polynomial of low-degree and low-weight. Roughly, each of these mechanisms works by maintaining a sequence of polynomials $p^{(1)}, p^{(2)}, \dots \in \mathcal{P}_{t,W}$ that give increasingly good approximations to the $\mathcal{Q}$-function family f_D. Moreover, the mechanism produces the next polynomial in the sequence by considering only one query $y^{(t)}$ that "distinguishes" the real database in the sense that $|p^{(t)}(y^{(t)}) - f_D(y^{(t)})|$ is large.

Syntactically, we will consider functions of the form $\mathbf{U} : \mathcal{P}_{t,W} \times \mathcal{Q} \times \mathbb{R} \to \mathcal{P}_{t,W}$. The inputs to $\mathbf{U}$ are a polynomial $p^{(t)} \in \mathcal{P}_{t,W}$, which is the current polynomial approximation; a query $y \in Y_{\mathcal{Q}}$, which represents the distinguishing query; and also a real number that estimates $f_D(y)$. Formally, we define a *database update sequence*, to capture the sequence of inputs to $\mathbf{U}$ used to generate the database sequence $p^{(1)}, p^{(2)}, \dots$.

Definition 3.1 (Database Update Sequence)**.** Let $D \in (\{0,1\}^d)^n$ be any database and let $\big\{(p^{(t)}, y^{(t)}, \widehat{a}^{(t)})\big\}_{t=1,\dots,C} \in (\mathcal{P}_{t,W} \times \mathcal{Q} \times \mathbb{R})^C$ be a sequence of tuples. We say the sequence is an $(\mathbf{U}, D, \mathcal{Q}, \alpha, C)$-*database update sequence* if it satisfies the following properties:

1. $p^{(1)} = D(\emptyset, \cdot, \cdot)$,
2. for every $t = 1, 2, \dots, C$, $\big|f_D(y^{(t)}) - p^{(t)}(y^{(t)})\big| \geq \alpha$,
3. for every $t = 1, 2, \dots, C$, $\big|f_D(y^{(t)}) - \widehat{a}^{(t)}\big| \leq \alpha/2$,
4. and for every $t = 1, 2, \dots, C-1$, $p^{(t+1)} = \mathbf{U}(p^{(t)}, y^{(t)}, \widehat{a}^{(t)})$.

We note that for all of the iterative database constructions we consider, the approximate answer $\widehat{a}^{(t)}$ is used only to determine the *sign* of $f_D(y^{(t)}) - p^{(t)}(y^{(t)})$, which is the motivation for requiring that $\widehat{a}^{(t)}$ have error smaller than α. The main measures of efficiency we're interested in from an iterative database construction are the maximum number of updates we need to perform before the database $p^{(t)}$ approximates D well with respect to the queries in $\mathcal{Q}$ and the time required to compute $\mathbf{U}$. To this end we define an iterative database construction as follows:

Definition 3.2 (Iterative Database Construction)**.** Let $\mathbf{U} : \mathcal{P}_{t,W} \times \mathcal{Q} \times \mathbb{R} \to \mathcal{P}_{t,W}$ be an update rule and let $B : \mathbb{R} \to \mathbb{R}$ be a function. We say $\mathbf{U}$ is a *$B(\alpha)$-iterative database construction for query class $\mathcal{Q}$* if for every database $D \in (\{0,1\}^d)^n$, every $(\mathbf{U}, D, \mathcal{Q}, \alpha, C)$-database update sequence satisfies $C \le B(\alpha)$.

Note that, by definition, if $\mathbf{U}$ is a $B(\alpha)$-iterative database construction, then given any maximal $(\mathbf{U}, D, \mathcal{Q}, \alpha, C)$-database update sequence, the final database $p^{(C)}$ must satisfy

$$\forall y \in Y_{\mathcal{Q}},\ \left| f_D(y) - p^{(C)}(y) \right| \le \alpha$$

or else there would exist another query satisfying property 2 of Definition 3.1, and thus there would exist a $(\mathbf{U}, D, \mathcal{Q}, \alpha, C+1)$-database update sequence, contradicting maximality.

Theorem 3.3 (Variant of [GRU12])**.** *For any $\alpha > 0$, and any family of linear queries $\mathcal{Q}$, if there is a $B(\alpha)$-iterative database construction, $\mathbf{U}$, for $\mathcal{Q}$ then there is an (ε, δ)-differentially private online algorithm that is $(4\alpha, \beta)$-accurate for any sequence of ℓ (possibly adaptively chosen) queries from $\mathcal{Q}$ so long as*

$$n \ge \frac{1000\sqrt{B(\alpha)}\log(\ell/\beta)\log(4/\delta)}{\alpha\varepsilon}.$$

Moreover, if $\mathbf{U}$ runs in time $T_{\mathbf{U}}$, then the private algorithm has running time $\mathrm{poly}(T_{\mathbf{U}})$ *per query.*

The IDC we will use is specified in Algorithm 1. We note that the algorithm will represent a polynomial as a vector $\overline{p}$ of length $2\binom{m}{\le t} + 1$ with only non-negative entries. For each coefficient $S \in \binom{[m]}{\le t}$, the vector will have two components $\overline{p}_S, \overline{p}_{\neg S}$. Intuitively these two entries represent the positive part and negative part of the coefficient c_S of p. There will also be an additional entry $\overline{p}_0$ that is used to ensure that the L_1-norm of the vector is exactly 1. Given a polynomial $p \in \mathcal{P}_{t,W}$ with coefficients (c_S), we can construct this vector by setting

$$\overline{p}_S = \frac{\max\{0, c_S\}}{W} \qquad \overline{p}_{\neg S} = \frac{\max\{0, -c_S\}}{W}$$

and choosing $\overline{p}_0$ so that $\|\overline{p}\|_1 = 1$. Observe that $\overline{p}_0$ can always be set appropriately since the weight of p is at most W.

We observe two things about $\overline{p}$: (1) Given a query $y \in \{-1,1\}^m$, we can construct a vector $\overline{y}$ of length $2\binom{m}{\le t} + 1$ in which $\overline{y}_0 = 0$, $\overline{y}_S = \prod_{i \in S} y_i$ and $\overline{y}_{\neg S} = -\prod_{i \in S} y_i$. This vector will satisfy $W\langle \overline{p}, \overline{y}\rangle = p(y)$. (2) $\overline{p}$ now represents a probability distribution on the $2\binom{m}{\le t}$ "coefficients."

We summarize the properties of the multiplicative weights algorithm in the following theorem:

Theorem 3.4. *For any $\alpha > 0$, and any family of linear queries $\mathcal{Q}$ if $\mathcal{P}_{t,W}$ $(\alpha/4)$-approximates the restriction $\mathcal{F}|_Y$ then Algorithm 1 is a $B(\alpha)$-iterative database construction, $\mathbf{U}$, for*

$$B(\alpha) = \frac{16W^2 \log\left(2\binom{m}{\le t} + 1\right)}{\alpha^2}.$$

Moreover, $\mathbf{U}$ runs in time $\mathrm{poly}\left(\binom{m}{\le t}\right)$.

Proof. Let $D \in (\{0,1\}^d)^n$ be any database and consider a $(\mathbf{U}^{MW}, D, \mathcal{Q}, \alpha, B)$-database update sequence, $\left\{(\overline{p}^{(t)}, y^{(t)}, \widehat{a}^{(t)})\right\}_{t=1,\ldots,B}$. It will be sufficient if we can show that $B \le 16W^2\log(2\binom{m}{\le t} + 1)/\alpha^2$. Specifically, that after $B = 16W^2\log(2\binom{m}{\le t} + 1)/\alpha^2$ invocations of $\mathbf{U}^{MW}$, the polynomial $\overline{p}^{(B)}$ is such that

$$\forall y \in Y_{\mathcal{Q}},\ |W\langle \overline{p}^{(B)}, \overline{y}\rangle - f_D(y)| \le \alpha.$$

Algorithm 1 The Multiplicative Weights Algorithm for Low-Weight Polynomials.

```
$\mathbf{U}_{\alpha}^{MW}(\overline{p}^{(t)}, y^{(t)}, \widehat{a}^{(t)})$:
  Let $\eta \leftarrow \alpha/4W$.
  If: $p^{(t)} = \emptyset$ then: output $\overline{p}^{(t)} = \frac{1}{2\binom{m}{\leq t}+1}(1, \dots, 1)$, a representation of the constant 0 polynomial.
  Else if: $\widehat{a}^{(t)} < W\langle \overline{p}^{(t)}, \overline{y}^{(t)} \rangle$
    Let $\overline{r}^{(t)} = \overline{y}^{(t)}$
  Else:
    Let $\overline{r}^{(t)} = -\overline{y}^{(t)}$
  Update: For all $S \in \binom{[m]}{\leq t}$ let
        $\overline{p}_S^{(t+1)} \leftarrow \exp(-\eta \overline{r}_S^{(t)}) \cdot \overline{p}_S^{(t)}, \qquad \overline{p}_{\neg S}^{(t+1)} \leftarrow \exp(-\eta \overline{r}_{\neg S}^{(t)}) \cdot \overline{p}_{\neg S}^{(t)}$
        $\overline{p}^{(t+1)} \leftarrow \frac{\overline{p}^{(t+1)}}{\|\overline{p}^{(t+1)}\|_1}$
  Output $\overline{p}^{(t+1)}$.
```

That is $\overline{p}^{(B)}$ represents a polynomial that approximates f_D.

First, we note that there always exists a polynomial $p_D \in \mathcal{P}_{t,W}$ such that

$$\forall y \in Y_{\mathcal{Q}}, \ |p_D(y) - f_D(y)| \leq \frac{\alpha}{4}. \tag{1}$$

The assumption of our theorem is that for every $x^{(i)} \in D$, there exists $p_{x^{(i)}} \in \mathcal{P}_{t,W}$ such that

$$\forall y \in Y_{\mathcal{Q}}, \ |p_{x^{(i)}}(y) - f_{x^{(i)}}(y)| \leq \frac{\alpha}{4}.$$

Thus, since $f_D = \frac{1}{n}\sum_{i=1}^{n} f_{x^{(i)}}$, the polynomial $p_D = \frac{1}{n}\sum_{i=1}^{n} p_{x^{(i)}}$ will satisfy (1). Note that $p_D \in \mathcal{P}_{t,W}$, thus if we represent p_D as a vector,

$$\forall y \in Y_{\mathcal{Q}}, \ |W\langle \overline{p}_D, \overline{y} \rangle - f_D(y)| \leq \frac{\alpha}{4}.$$

Given the existence of $\overline{p}_D$, we will define a potential function capturing how far $\overline{p}^{(t)}$ is from $\overline{p}_D$. Specifically, we define

$$\Psi_t := KL(\overline{p}_D || \overline{p}^{(t)}) = \sum_{I \in \left\{0, \binom{[m]}{\leq t}, \neg\binom{[m]}{\leq t}\right\}} \overline{p}_{D,I} \log\left(\frac{\overline{p}_{D,I}}{\overline{p}_I^{(t)}}\right)$$

to be the KL divergence between $\overline{p}_D$ and the current approximation $\overline{p}^{(t)}$. Note that the sum iterates over all $2\binom{m}{\leq t} + 1$ indices in $\overline{p}$. We have the following fact about KL divergence.

Fact 3.5. *For all t: $\Psi_t \geq 0$, and $\Psi_0 \leq \log\left(2\binom{m}{\leq t} + 1\right)$.*

We will argue that in each step the potential drops by at least $\alpha^2/16W^2$. Because the potential begins at $\log\left(2\binom{m}{\leq t} + 1\right)$, and must always be non-negative, we know that there can be at most $B(\alpha) \leq 16W^2 \log\left(2\binom{m}{\leq t} + 1\right)/\alpha^2$ steps before the algorithm outputs a (vector representation of) a polynomial that approximates f_D on $Y_{\mathcal{Q}}$.

Lemma 3.6 (Modification of [HR10])**.**

$$\Psi_t - \Psi_{t+1} \geq \eta \left(\langle \overline{p}^{(t)}, r^{(t)} \rangle - \langle \overline{p}_D, r^{(t)} \rangle \right) - \eta^2$$

Proof.

$$\begin{aligned}
\Psi_t - \Psi_{t+1} &= \sum_{I \in \left\{0, \binom{[m]}{\leq t}, \neg\binom{[m]}{\leq t}\right\}} \overline{p}_{D,I} \log \left(\frac{\overline{p}_I^{(t+1)}}{\overline{p}_I^{(t)}} \right) \\
&= -\eta \langle \overline{p}_D, r^{(t)} \rangle - \log \left(\sum_{I \in \left\{0, \binom{[m]}{\leq t}, \neg\binom{[m]}{\leq t}\right\}} \exp(-\eta r_I^{(t)}) \overline{p}_I^{(t)} \right) \\
&\geq -\eta \langle \overline{p}_D, r^{(t)} \rangle - \log \left(\sum_{I \in \left\{0, \binom{[m]}{\leq t}, \neg\binom{[m]}{\leq t}\right\}} \overline{p}_I^{(t)} (1 + \eta^2 - \eta r_I^{(t)}) \right) \\
&\geq \eta \left(\langle \overline{p}^{(t)}, r^{(t)} \rangle - \langle \overline{p}_D, r^{(t)} \rangle \right) - \eta^2
\end{aligned}$$

□

The rest of the proof now follows easily. By the conditions of an iterative database construction algorithm, $|\widehat{a}^{(t)} - f_D(y^{(t)})| \leq \alpha/2$. Hence, for each t such that $|W \langle \overline{p}^{(t)}, y^{(t)} \rangle - f_D(y^{(t)})| \geq \alpha$, we also have that $W \langle \overline{p}^{(t)}, y^{(t)} \rangle > f_D(y^{(t)})$ if and only if $W \langle \overline{p}^{(t)}, y^{(t)} \rangle > \widehat{a}^{(t)}$.

In particular, if $r^{(t)} = y^{(t)}$, then $W \langle \overline{p}^{(t)}, y^{(t)} \rangle - W \langle \overline{p}_D, y^{(t)} \rangle \geq \alpha/2$. Similarly, if $r^{(t)} = -y^{(t)}$, then $W \langle \overline{p}_D, y^{(t)} \rangle - W \langle \overline{p}^{(t)}, y^{(t)} \rangle \geq \alpha$. Here we have utilized the fact that $|p_D(y) - f_D(y)| \leq \alpha/4$. Therefore, by Lemma 3.6 and the fact that $\eta = \alpha/4W$:

$$\Psi_t - \Psi_{t+1} \geq \frac{\alpha}{4W} \left(\langle \overline{p}^{(t)}, r^{(t)} \rangle - \langle \overline{p}_D, r^{(t)} \rangle \right) - \frac{\alpha^2}{16W^2} \geq \frac{\alpha}{4W} \left(\frac{\alpha}{2W} \right) - \frac{\alpha^2}{16W^2} = \frac{\alpha^2}{16W^2}$$

□

4 Upper Bounds

Fact 2.5 shows that in order to develop a differentially private mechanism that can release all k-way marginals of a database, it is sufficient to construct low-weight polynomials that approximate OR_d, the d-variate OR function, on all Boolean inputs of Hamming weight at most k. This is the purpose to which we now turn.

The OR_d function is easily seen to have an exact polynomial representation of constant weight and degree d (e.g. see Fact 4.3 below); however, an approximation with smaller degree may be achieved at the expense of larger weight. The best known weight-degree tradeoff, implicit in the work of Servedio et al. [STT12], can be stated as follows: there exists a polynomial p of degree t and weight $(d \log(1/\gamma)/t)^{(d(\log 1/\gamma)^2/t)}$ that γ-approximates the OR_d function on all Boolean inputs, for every t larger than $\sqrt{d} \log(1/\gamma)$. Setting the degree t to be $O(d/\log^{.99} d)$ yields a polynomial of weight at most $d^{0.01}$ that approximates the OR_d function over the entire Boolean hypercube to any desired constant accuracy. On the other hand, Lemma 8 of [STT12] can be shown to imply that any polynomial of weight W that 1/3-approximates the OR_d function requires degree $\Omega(d/\log W)$, essentially matching the $O(d/\log^{.99} d)$ upper bound of Servedio et al. when $W = d^{\Omega(1)}$.

However, in order to privately release k-way marginals, we have shown that it suffices to construct polynomials that are accurate *only* on inputs of low Hamming weight. In this section, we give a construction that achieves significantly improved weight degree trade-offs in this setting. In the next section, we demonstrate the tightness of our construction by proving matching lower bounds.

We construct our approximations by decomposing the d-variate OR function into an OR of OR's, which is the same approach taken by Servedio et al. [STT12]. Here, the outer OR has fan-in m and the inner OR has fan-in d/m, where the subsequent analysis will determine the appropriate choice of m. In order to obtain an approximation that is accurate on all Boolean inputs, Servedio et al. approximate the outer OR using a transformation of the Chebyshev polynomials of degree $\sqrt{m}$, and compute each of the inner OR's exactly.

For $k \ll \log^2 d$, we are able to substantially reduce the degree of the approximating polynomial, relative to the construction of Servedio et al., by leveraging the fact that we are interested in approximations that are accurate only on inputs of Hamming weight at most k. Specifically, we are able to approximate the outer OR function using a polynomial of degree only $\sqrt{k}$ rather than $\sqrt{m}$, and argue that the weight of the resulting polynomial is still bounded by a polynomial in d.

We now proceed to prove the main lemmas. For the sake of intuition, we begin with weight-degree tradeoffs in the simpler setting in which we are concerned with approximating the OR_d function over the entire Boolean hypercube. The following lemma, proved below for completeness, is implicit in the work of [STT12].

Lemma 4.1. *For every $\gamma > 0$ and $m \in [d]$, there is a polynomial of degree $t = O(d \log(1/\gamma)/\sqrt{m})$ and weight $W = m^{O(\sqrt{m}\log(1/\gamma)}$ that γ-approximates the OR_d function.*

Our main contribution in this section is the following lemma that gives an improved polynomial approximation to the OR_d function restricted to inputs of low Hamming weight.

Lemma 4.2. *For every $\gamma > 0$, $k < d$ and $m \in [d] \setminus [k]$, there is a polynomial of degree $t = O(d\sqrt{k}\log(1/\gamma)/m)$ and weight $W = m^{O(\sqrt{k}\log(1/\gamma))}$ that γ-approximates the OR_d function restricted to inputs of Hamming weight at most k.*

For any constant accuracy, one may take $m = d^{O(1/\sqrt{k})}$ in the lemma (here the choice of constant depends on the constants in Fact 4.4 and the desired accuracy) and obtain a polynomial of degree $d^{1-\Omega(1/\sqrt{k})}$ and weight $d^{.01}$.

Our constructions use the following basic facts.

Fact 4.3. *The real polynomial $p_d : \{1, -1\}^d \to \mathbb{R}$*

$$p_d(x) = 2\left(\sum_{S \subseteq [d]} 2^{-d} \prod_{i \in S} x_i\right) - 1 = 2\prod_{i=1}^{d}\left(\frac{1+x_i}{2}\right) - 1$$

computes $OR_d(x)$ and has weight $w(p_d) \leq 3$.

Fact 4.4. *[see e.g. [TUV12]] For every $k \in \mathbb{N}$ and $\gamma > 0$, there exists a univariate real polynomial $p = \sum_{i=0}^{t_k} c_i x^i$ of degree t_k such that*

1. *$t_k = O(\sqrt{k}\log(1/\gamma))$,*

2. *for every $i \in [t_k]$, $|c_i| \leq 2^{O(\sqrt{k}\log(1/\gamma))}$,*

3. *$p(0) = 0$, and*

4. for every $x \in [2k]$, $|p(x) - 1| \leq \gamma/2$.

Proof of Lemma 4.1. We can compute $\mathrm{OR}_d(y)$ as a disjunction of disjunctions by partitioning the inputs $y_1, \ldots, y_d$ into blocks of size d/m and computing:

$$\mathrm{OR}_m(\mathrm{OR}_{d/m}(y_1, \ldots, y_{d/m}), \ldots, \mathrm{OR}_{d/m}(y_{d-d/m+1}, \ldots, y_d)).$$

In order to approximately compute $\mathrm{OR}_d(y)$, we compute the inner disjunctions exactly using the polynomial $p_{d/m}$ given in Fact 4.3 and approximate the outer disjunction using the polynomial from Fact 4.4. Let

$$Z(y) = p_{d/m}(y_1, \ldots, y_{d/m}) + \cdots + p_{d/m}(y_{d-d/m+1}, \ldots, y_d).$$

Setting $k = m$ in Fact 4.4, let q_m be the resulting polynomial of degree $O(\sqrt{m}\log(1/\gamma))$ and weight $O(m^{\sqrt{m}\log(1/\gamma)})$. Our final polynomial is

$$1 - 2q_m(m - Z(y)).$$

Note that $m - Z(y)$ takes values in $\{0, \ldots, m\}$ and is 0 exactly when all inputs $y_1, \ldots, y_d$ are FALSE. It follows that our final polynomial indeed approximates OR_d to additive error γ on all Boolean inputs.

We bound the degree and weight of this polynomial in y. By Fact 4.3, the inner disjunctions are computed exactly using degree d/m and weight at most 3. Hence, the total degree is $O(\sqrt{m}\log(1/\gamma) \cdot d/m)$. To bound the weight, we observe that the outer polynomial $q_m(\cdot)$ has at most $T = m^{O(\sqrt{m}\log(1/\gamma))}$ terms where each one has degree at most $D_{\text{outer}} = O(\sqrt{m}\log(1/\gamma))$ and coefficients of absolute value at most $c_{\text{outer}} = 2^{O(\sqrt{m}\log(1/\gamma))}$. Expanding the polynomials for $Z(y)$, the weight of each term incurs a multiplicative factor of $c_{\text{inner}} \leq 3^{D_{\text{outer}}} = 3^{O(\sqrt{m}\log 1/\gamma)}$ so the total weight is at most $c_{\text{inner}} \cdot c_{\text{outer}} \cdot T = m^{O(\sqrt{m}\log 1/\gamma)}$.

□

Proof of Lemma 4.2. Again we partition the inputs $y_1, \ldots, y_d$ into blocks of size d/m and view the disjunction as:

$$\mathrm{OR}_m(\mathrm{OR}_{d/m}(y_1, \ldots, y_{d/m}), \ldots, \mathrm{OR}_{d/m}(y_{d-d/m+1}, \ldots, y_d)).$$

Once again, we compute the inner disjunctions exactly using the polynomial from Fact 4.3. Let

$$Z(y) = p_{d/m}(y_1, \ldots, y_{d/m}) + \cdots + p_{d/m}(y_{d-d/m+1}, \ldots, y_d).$$

If the input y has Hamming weight at most k, then $Z(y)$ also takes values in $\{m, \ldots, m - 2k\}$. Thus, we may approximate the outer disjunction using a polynomial of degree $O(\sqrt{k}\log(1/\gamma))$ from Fact 4.4. Our final polynomial is:

$$1 - 2q_k(m - Z).$$

The bound on degree and weight may be obtained as in the previous lemma. □

4.1 Proof of Theorem 1.1

We now present the proof of Theorem 1.1.

Proof of Theorem 1.1. Taking $m = O\big((\log d/\log\log d)^2\big)$ in Lemma 4.1, taking $m = d^{O(1/\sqrt{k})}$ in Lemma 4.2, and combining with Fact 2.5, it follows that for some constant $C > 0$, the family of d-variate disjunctions restricted to $H_{d,k}$ is 0.01-approximated by the family of d-variate real polynomials of degree t and weight W where

$$t = \min\left\{d^{1-\frac{1}{C\sqrt{k}}}, \frac{d}{\log^{0.995} d}\right\} \text{ and } W = d^{0.01}.$$

Consequently, by Theorem 3.4, we have an algorithm that is a $B(1/400)$-iterative database construction where

$$B(1/400) = O(d^{0.02} t \log d) = O\left(d^{0.02}\log d \cdot \min\left\{d^{1-\frac{1}{C\sqrt{k}}}, \frac{d}{\log^{0.995} d}\right\}\right)$$

and the algorithm runs in time $T = \mathrm{poly}(\binom{d}{\leq t})$

Thus, by Theorem 3.3, we have an (ε, δ)-differentially private online algorithm that is $(0.01, 0.01)$-accurate for any sequence $\mathcal{Q}$ of (possibly adaptively chosen) k-way marginal queries provided the size of the database

$$n = \Omega\left(\left(\frac{1}{\varepsilon}\log(100|\mathcal{Q}|)\log\left(\frac{1}{\delta}\right)\right) d^{0.01}\sqrt{\log d} \cdot \min\left\{d^{0.5-\frac{1}{2C\sqrt{k}}}, \frac{d^{0.5}}{\log^{0.498} d}\right\}\right).$$

Further, the algorithm runs in time $\mathrm{poly}(T) = \mathrm{poly}(\binom{d}{\leq t}) = \min\left\{\exp\left(d^{1-1/C\sqrt{k}}\right), \exp\left(d/\log^{0.99} d\right)\right\}$.

□

Remark 1 in the Introduction follows from using a slightly different choice of m in Lemma 4.1, namely $m = O(\log^2 d/\log^3 \log d)$.

To obtain the summary of the database promised in Remark 2, we request an answer to each of the k-way marginal queries $B(1/400)$ times. Doing so, will ensure that we obtain a maximal database update sequence, and it was argued in Section 2.1 that the polynomial resulting from any maximal database update sequence accurately answers every k-way marginal query. Finally, we obtain a compact summary by randomly choosing $\tilde{O}(kd^{.01})$ samples from the normalized coefficient vector of this polynomial to obtain a new sparse polynomial that accurately answers every k-way marginal query (see e.g. [BS92]). Our compact summary is this final sparse polynomial.

5 Lower Bounds

In this section, we address the general problem of approximating a block-composed function $G = F(\ldots, f(.), \ldots)$, where $F\colon \{-1,1\}^k \to \{-1,1\}$, $f\colon Y \to \{-1,1\}$, $Y \subseteq \mathbb{R}^{d/k}$ over inputs restricted to a set $\mathcal{Y} \subseteq Y^k$ using low-weight polynomials. We give a lower bound on the minimum degree of such polynomials. In our main application, G will equal OR_d, and $\mathcal{Y}$ will be the set of all length d Boolean vectors of Hamming weight at most k.

Our proof technique is inspired by the composition theorem lower bounds shown in [She09, Theorem 3.1], where it is shown that the γ-approximate degree of the composed function G is at least the product of the γ-approximate degree of the outer function and the PTF degree of the inner function. Our main contribution is a generalization of such a composition theorem along two directions: (1) we show degree lower bounds that take into account the L1-norm of the coefficient vector of the approximating polynomial, and (2) our lower bounds hold even when we only require the approximation to be accurate on inputs of low Hamming weight, while prior work only considered approximations that are accurate on the entire Boolean hypercube.

Our main theorem is stated below. In parsing the statement of the theorem, it may be helpful to think of $G = \mathrm{OR}_d$, $\mathcal{Y} = H_{d,k}$, the set of all length d Boolean vectors of Hamming weight at most k, $f = \mathrm{OR}_{d/k}$, $F = \mathrm{OR}_k$, $Y = \{-1,1\}^{d/k}$, and $H = H_{d/k,1}$, the set of all Boolean vectors of Hamming weight at most 1. This will be the setting of interest in our main application of the theorem.

Theorem 5.1. *Let $Y \subset \mathbb{R}^{d/k}$ be a finite set and $\gamma > 0$. Given $f : Y \to \{-1,1\}$ and $F : \{-1,1\}^k \to \{-1,1\}$ such that $deg_{2\gamma}(F) = D$, let $G : Y^k \to \{-1,1\}$ denote the composed function defined by $G(Y_1,\ldots,Y_k) = F(f(Y_1),\ldots,f(Y_k))$. Let $\mathcal{Y} \subseteq Y^k$. Suppose there exists $H \subseteq Y$ such that for every $(Y_1,\ldots,Y_k) \in Y^k \setminus \mathcal{Y}$ there exists $i \in [k]$ such that $Y_i \in Y \setminus H$. Then, for every $t \in \mathbb{Z}_+$,*

$$deg_{(\gamma,W)}(G|_{\mathcal{Y}}) \geq \frac{1}{2}tD \text{ for every } W \leq \gamma 2^{-k} w^*(f|_H, t)^{\frac{D}{2}}.$$

We derive the following corollary from Theorem 5.1. Theorem 1.3 follows immediately from Corollary 5.2 by considering any $k = o(\log d)$.

Corollary 5.2. *Let $k \in [d]$. Then, there exists a universal constant $C > 0$ such that*

$$deg_{(1/6,W)}(OR_d|_{H_{d,k}}) = \Omega\left(\frac{d}{\sqrt{k}} \cdot \frac{W^{-\frac{1}{C\sqrt{k}}}}{2^{\sqrt{k}/C}}\right).$$

Intuition underlying our proof technique. Recall that our upper bound in Section 4 worked as follows. We viewed OR_d as an "OR of ORs", and we approximated the outer OR with a polynomial p of degree $\deg_{\text{outer}}$ chosen to be as small as possible, and composed p with a low-weight but high-degree polynomial computing each inner OR. We needed to make sure the weight W_{inner} of the inner polynomials was very low, because the composition step potentially blows the weight up to roughly $W_{\text{inner}}^{\deg_{\text{outer}}}$. As a result, the inner polynomials had to have very high degree, to keep their weight low.

Intuitively, we construct a dual solution to a certain linear program that captures the intuition that any low-weight, low-degree polynomial approximation to OR_d must look something like our primal solution, composing a low-degree approximation to an "outer" OR with low-weight approximations to inner ORs. Moreover, our dual solution formalizes the intuition that the composition step must result in a massive blowup in weight, from W_{inner} to roughly $W_{\text{inner}}^{\deg_{\text{outer}}}$.

In more detail, our dual construction works by writing OR_d as an OR of ORs, where the outer OR is over k variables, and each inner ORs is over d/k variables. We obtain our dual solution by carefully combining a dual witness Γ to the high approximate degree of the outer OR, with a dual witness ψ to the fact that any low-degree polynomial with margin at least 1 for each inner OR, must have "large" weight, even if the polynomial must satisfy the margin constraint only on inputs of Hamming weight 0 or 1. This latter condition, that ψ must witness high non-constant margin weight even if restricted to inputs of Hamming weight 0 or 1, is essential to ensuring that our combined dual witness does not place any "mass" on irrelevant inputs, i.e. those of Hamming weight larger than k.

5.1 Duality Theorems

In the rest of the section, we let $\chi_S(x) = \prod_{i\in S} x_i$ for any given set $S \subseteq [d]$. The question of existence of a weight W polynomial with small degree that γ-approximates a given function can be expressed as a feasibility problem for a linear program. Now, in order to show the non-existence of such a polynomial, it is sufficient to show infeasibility of the linear program. By duality, this is equivalent to demonstrating existence of a solution to the corresponding dual program. We begin by summarizing the duality theorems that will be useful in exhibiting this witness.

Theorem 5.3 (Duality Theorem for (γ, W)-approximate degree)**.** *Fix $\gamma \geq 0$ and let $f : Y \to \{-1, 1\}$ be given for some finite set $Y \subset \mathbb{R}^d$. Then, $deg_{(\gamma,W)}(f) \geq t+1$ if and only if there exists a function $\Psi : Y \to \mathbb{R}$ such that*

1. $\sum_{y \in Y} |\Psi(y)| = 1$,
2. $\sum_{y \in Y} \Psi(y) f(y) - W \cdot |\sum_{y \in Y} \Psi(y)\chi_S(y)| > \gamma$ *for every* $S \subseteq [d], |S| \leq t$.

Proof. By definition, $\deg_{(\gamma,W)}(f) \leq t$ if and only if $\exists (\lambda_S)_{S \subseteq [d], |S| \leq t}$:

$$\sum_{S \subseteq [d], |S| \leq t} |\lambda_S| \leq W, \text{ and}$$

$$\left| f(y) - \sum_{S \subseteq [d], |S| \leq t} \lambda_S \chi_S(y) \right| \leq \gamma \; \forall \; y \in Y.$$

By Farkas' lemma, $\deg_{(\gamma,W)}(f) \leq t$ if and only if $\nexists \; \Psi : Y \to \mathbb{R}$ such that

$$\frac{1}{W} \sum_{y \in Y} (f(y)\Psi(y) - \gamma|\Psi(y)|) > \left| \sum_{y \in Y} \chi_S(y)\Psi(y) \right| \quad \forall \; S \subseteq [d], |S| \leq t.$$

□

The dual witness that we construct to prove Theorem 5.1 is obtained by combining a dual witness for the large non-constant margin weight of the inner function with a dual witness for the large approximate degree for the outer function. The duality conditions for these are given below. The proof of the duality condition for the case of γ-approximate degree is well-known, and we omit the proof for brevity (see e.g. [She11, Š08, BT13]).

Theorem 5.4 (Duality Theorem for γ-approximate degree)**.** *Fix $\gamma \geq 0$ and let $f : Y \to \{-1, 1\}$ be given, where $Y \subset \mathbb{R}^d$ is a finite set. Then, $deg_\gamma(f) \geq t + 1$ if and only if there exists a function $\Gamma : Y \to \mathbb{R}$ such that*

1. $\sum_{y \in Y} |\Gamma(y)| = 1$,
2. $\sum_{y \in Y} \Gamma(y)p(y) = 0$ *for every polynomial p of degree at most t, and*
3. $\sum_{y \in Y} \Gamma(y) f(y) > \gamma$.

Theorem 5.5 (Duality Theorem for non-constant margin weight)**.** *Let $Y \subset \mathbb{R}^d$ be a finite set, let $f : Y \to \{1, -1\}$ be a given function and $w > 0$. The non-constant margin weight $w^*(f, t) \geq w$ if and only if there exists a distribution $\mu : Y \to [0, 1]$ such that*

1. $\sum_{y \in Y} \mu(y) f(y) = 0$
2. $\left| \sum_{y \in Y} \mu(y) f(y) \chi_S(y) \right| \leq \frac{1}{w}$ *for every* $S \subseteq [d]$, $|S| \leq t$.

Proof. Let $\mathcal{S} = \{S \subseteq [d] : |S| \leq t\}$, $\overline{\mathcal{S}} = \mathcal{S} \setminus \{\emptyset\}$. By definition, $w^*(f, t)$ is expressed by the following linear program:

$$\min \sum_{S \in \overline{\mathcal{S}}} |\lambda_S|$$

$$f(y) \sum_{S \in \mathcal{S}} \lambda_S \chi_S(y) \geq 1 \; \forall \; y \in Y.$$

The above linear program can be restated as follows:

$$\begin{aligned} \min \sum_{S\in\overline{\mathcal{S}}} \alpha_S & \\ \alpha_S + \lambda_S \geq 0 \ \forall \ S \in \overline{\mathcal{S}}, & \\ \alpha_S - \lambda_S \geq 0 \ \forall \ S \in \overline{\mathcal{S}}, & \\ f(y)\sum_{S\in\mathcal{S}} \lambda_S \chi_S(y) \geq 1 \ \forall \ y \in Y, \text{ and} & \\ \alpha_S \geq 0 \ \forall \ S \in \overline{\mathcal{S}}. & \end{aligned}$$

The dual program is expressed below:

$$\begin{aligned} \max \sum_{y} \mu(y) & \\ u_1(S) + u_2(S) \leq 1 \ \forall \ S \in \overline{\mathcal{S}}, & \\ \sum_{y\in Y} \mu(y) f(y) \chi_S(y) + u_1(S) - u_2(S) = 0 \ \forall \ S \in \overline{\mathcal{S}}, & \\ \sum_{y\in Y} \mu(y) f(y) = 0, & \\ \mu(y) \geq 0 \ \forall \ y \in Y, \ u_1(S), u_2(S) \geq 0 \ \forall \ S \in \overline{\mathcal{S}}. & \end{aligned}$$

By standard manipulations, the above dual program is equivalent to

$$\begin{aligned} \max \sum_{y} \mu(y) & \\ |\sum_{y\in Y} \mu(y) \chi_S(y) f(y)| \leq 1 \ \forall \ S \in \overline{\mathcal{S}} & \\ \sum_{y\in Y} \mu(y) f(y) = 0, & \\ \mu(y) \geq 0 \ \forall \ y \in Y & \end{aligned}$$

Finally, given a distribution μ' satisfying the hypothesis of the theorem, one can obtain a dual solution μ to show that $w^*(f,t) \geq w$ by taking $w^{-1} = \max_{S\in\mathcal{S}} |\sum_{y\in Y} \mu'(y)\chi_S(y)f(y)|$ and setting $\mu(y) = w\mu'(y) \ \forall \ y \in Y$. In the other direction, if $w^*(f,t) \geq w$, then we have a dual solution μ satisfying the above dual program such that $\sum_{y\in Y} \mu(y) = w^*(f,t)$. By setting $\mu'(y) = \mu(y)/w^*(f,t) \ \forall \ y \in Y$, we obtain the desired distribution. □

5.2 Proof of Theorem 5.1

Our approach to exhibiting a dual witness as per Theorem 5.3 is to build a dual witness by appropriately combining the dual witnesses for the "hardness" of the inner and outer functions. Our method of combining the dual witnesses is inspired by the technique of [She09, Theorem 3.7].

Proof of Theorem 5.1. Let $w = w^*(f|_H, t)$. We will exhibit a dual witness function $\Psi : \mathcal{Y} \to \mathbb{R}$ corresponding to Theorem 5.3 for the specified choice of degree and weight. For $y \in Y^k$, let

$Y_i = (y_{(i-1)(d/k)+1}, \ldots, y_{id/k})$. By Theorem 5.5, we know that there exists a distribution $\mu : H \to \mathbb{R}$ such that

$$\begin{aligned} \sum_{y \in H} \mu(y) f(y) &= 0, && (2) \\ \left| \sum_{y \in H} \mu(y) f(y) \chi_S(y) \right| &\leq \frac{1}{w} \; \forall\, S \subseteq \left[\frac{d}{k}\right], |S| \leq t && (3) \end{aligned}$$

We set $\mu(y) = 0$ for $y \in Y \setminus H$.

Since $\deg_{2\gamma}(F) = D$, by Theorem 5.4, we know that there exists a function $\Gamma : \{-1,1\}^k \to \mathbb{R}$ such that

$$\begin{aligned} \sum_{x \in \{-1,1\}^k} |\Gamma(x)| &= 1, && (4) \\ \sum_{x \in \{-1,1\}^k} \Gamma(x) p(x) &= 0 \text{ for every polynomial } p \text{ of degree at most } D, \text{ and} && (5) \\ \sum_{x \in \{-1,1\}^k} \Gamma(x) F(x) &> 2\gamma. && (6) \end{aligned}$$

Consider the function $\Psi : Y^k \to \mathbb{R}$ defined as $\Psi(y) = 2^k \Gamma(f(Y_1), \ldots, f(Y_k)) \prod_{i=1}^k \mu(Y_i)$. By the hypothesis of the theorem, we know that if $(Y_1, \ldots, Y_k) \in Y^k \setminus \mathcal{Y}$, then there exists $i \in [k]$ such that $Y_i \in Y \setminus H$ and hence $\mu(Y_i) = 0$ and therefore $\Psi(Y_1, \ldots, Y_k) = 0$.

1.
$$\begin{aligned} \sum_{y \in \mathcal{Y}} |\Psi(y)| &= \sum_{y \in \mathcal{Y}} 2^k |\Gamma(f(Y_1), \ldots, f(Y_k))| \prod_{i=1}^k \mu(Y_i) \\ &= 2^k \mathbb{E}_{y \sim \Phi}(|\Gamma(f(Y_1), \ldots, f(Y_k))|) \end{aligned}$$

   where $y \sim \Phi$ denotes y chosen from the product distribution $\Phi : Y^k \to [0,1]$ defined by $\Phi(y) = \prod_{i \in [k]} \mu(Y_i)$. Since $\sum_{y \in Y} \mu(y) f(y) = 0$, it follows that if Y_i is chosen with probability $\mu(Y_i)$, then $f(Y_i)$ is uniformly distributed in $\{-1,1\}$. Consequently,

$$\sum_{y \in \mathcal{Y}} |\Psi(y)| = 2^k \mathbb{E}_{z \sim_U \{-1,1\}^k}(|\Gamma(z_1, \ldots, z_k)|) = 1.$$

   The last equality is by using (4).

2. By the same reasoning as above, it follows from (6) that

$$\sum_{y \in \mathcal{Y}} \Psi(y) G(y) = \sum_{z \in \{-1,1\}^k} \Gamma(z) F(z) > 2\gamma.$$

3. Fix a subset $S \subseteq [d]$ of size at most $tD/2$. Let $S_i = S \cap \{(i-1)(d/k)+1, \ldots, id/k\}$ for each $i \in [k]$. Consequently, $\chi_S(y) = \prod_{i=1}^k \chi_{S_i}(Y_i)$.

   Now using the Fourier coefficients $\hat{\Gamma}(T)$ of the function Γ, we can express

$$\Gamma(z_1, \ldots, z_k) = \sum_{T \subseteq [k]} \hat{\Gamma}(T) \prod_{i \in T} z_i = \sum_{\substack{T \subseteq [k], \\ |T| \geq D}} \hat{\Gamma}(T) \prod_{i \in T} z_i$$

since $\hat{\Gamma}(T) = 0$ if $|T| < D$ by (5). Hence,

$$\Psi(y) = 2^k \sum_{\substack{T\subseteq[k],\\ |T|\geq D}} \hat{\Gamma}(T) \prod_{i\in T} f(Y_i)\mu(Y_i) \cdot \prod_{i\in[k]\setminus T} \mu(Y_i)$$

Therefore, $\sum_{y\in\mathcal{Y}} \Psi(y)\chi_S(y)$

$$\begin{aligned}
&= \sum_{y\in\mathcal{Y}} \Psi(y) \prod_{i\in[k]} \chi_{S_i}(Y_i) \\
&= 2^k \sum_{y\in\mathcal{Y}} \left(\sum_{\substack{T\subseteq[k],\\ |T|\geq D}} \hat{\Gamma}(T) \prod_{i\in T} f(Y_i)\mu(Y_i) \cdot \prod_{i\in[k]\setminus T} \mu(Y_i) \right) \prod_{i\in[k]} \chi_{S_i}(Y_i) \\
&= 2^k \sum_{\substack{T\subseteq[k],\\ |T|\geq D}} \hat{\Gamma}(T) \sum_{y\in\mathcal{Y}} \left(\prod_{i\in T} f(Y_i)\mu(Y_i) \cdot \prod_{i\in[k]\setminus T} \mu(Y_i) \prod_{i\in[k]} \chi_{S_i}(Y_i) \right) \\
&= 2^k \sum_{\substack{T\subseteq[k],\\ |T|\geq D}} \hat{\Gamma}(T) \sum_{Y_1,\ldots,Y_k\in H} \left(\prod_{i\in T} f(Y_i)\mu(Y_i)\chi_{S_i}(Y_i) \cdot \prod_{i\in[k]\setminus T} \mu(Y_i)\chi_{S_i}(Y_i) \right).
\end{aligned}$$

Rearranging, we have $\sum_{y\in\mathcal{Y}} \Psi(y)\chi_S(y) =$

$$2^k \sum_{\substack{T\subseteq[k],\\ |T|\geq D}} \hat{\Gamma}(T) \prod_{i\in T} \left(\sum_{Y_i\in H} f(Y_i)\mu(Y_i)\chi_{S_i}(Y_i) \right) \prod_{i\in[k]\setminus T} \left(\sum_{Y_i\in H} \mu(Y_i)\chi_{S_i}(Y_i) \right). \tag{7}$$

Now, we will bound each product term in the outer sum by $w^{-D/2}$. We first observe that for every $i \in [k]$,

$$\sum_{x\in H} \mu(x)\chi_{S_i}(x) \leq \sum_{x\in H} \mu(x) = 1.$$

If $|S_i| \leq t$, by (3)

$$\left| \sum_{x\in H} f(x)\mu(x)\chi_{S_i}(x) \right| \leq \frac{1}{w}.$$

If $|S_i| > t$, then

$$\left| \sum_{x\in H} f(x)\mu(x)\chi_{S_i}(x) \right| \leq \sum_{x\in H} \mu(x) = 1.$$

Since $\sum_{i=1}^{k} |S_i| \leq tD/2$, it follows that $|S_i| \leq t$ for more than $k - D/2$ indices $i \in [k]$. Thus, for each $T \subseteq [k]$ such that $|T| \geq D$, there are at least $D/2$ indices $i \in T$ such that $|S_i| \leq t$. Hence,

$$\left| \sum_{y\in\mathcal{Y}} \Psi(y)\chi_S(y) \right| \leq 2^k w^{-\frac{D}{2}} \sum_{\substack{T\subseteq[k],\\ |T|\geq D}} \left| \hat{\Gamma}(T) \right| \leq 2^k w^{-\frac{D}{2}}.$$

Here, the last inequality is because $|\hat{\Gamma}(T)| \leq 2^{-k}$ from (4).

From 1, 2 and 3, we have

$$\sum_{y\in\mathcal{Y}} \Psi(y)G(y) - W \max_{S\subseteq[d],|S|\leq\frac{tD}{2}} \left|\sum_{y\in\mathcal{Y}} \Psi(y)\chi_S(y)\right| > \gamma$$

if $W \leq \gamma 2^{-k} w^{D/2}$. □

We now derive Corollary 5.2. We need the following theorems on the approximate degree and the non-constant margin weight of the OR_d function.

Theorem 5.6 (Approximate degree of OR_d)**.** *[Pat92] $deg_{1/3}(OR_d) = \Theta(\sqrt{d})$.*

Lemma 5.7 (Non-constant margin weight of OR_d)**.** $w^*(OR_d|_{H_{d,1}}, t) \geq d/t$.

Proof. The function

$$\mu(x) = \begin{cases} 1/2 & \text{if } x = (1, \dots, 1), \\ 1/2d & \text{if } x \in H_{d,1} \setminus \{(1, \dots, 1)\}. \end{cases}$$

acts as the dual witness in Theorem 5.5. □

Proof of Corollary 5.2. We use Theorem 5.1 in the following setting. Let $Y = \{-1,1\}^{d/k}$, the inner function $f : Y \to \{-1, 1\}$ be $\mathrm{OR}_{d/k}$ and the outer function $F : \{-1, 1\}^k \to \{-1, 1\}$ be OR_k, $\mathcal{Y} = H_{d,k}$ and $H = H_{d/k,1}$. By a simple counting argument, if $(Y_1, \dots, Y_k) \in \{-1, 1\}^d \setminus H_{d,k}$, then there exists $i \in [k]$ such that $Y_i \in \{-1, 1\}^{d/k} \setminus H_{d/k,1}$. Further, by Theorem 5.6, we know that $\deg_{1/3}(F) = \Theta(\sqrt{k})$ and by Claim 5.7, we know that $w^*(f|_H, t) \geq d/kt$. Therefore, by Theorem 5.1, we have that, for every $t \in \mathbb{Z}_+$,

$$\deg_{1/6,W}(OR_d|_{H_{d,k}}) = \Omega\left(t\sqrt{k}\right) \text{ for every } W \leq \frac{1}{6}2^{-k}\left(\frac{d}{kt}\right)^{C\sqrt{k}}.$$

We obtain the conclusion by taking $t = \lfloor (d/k)(6W2^k)^{-1/C\sqrt{k}} \rfloor$. □

Comparison to [STT12]. As described in the beginning of Section 4, Lemma 8 of the work of Servedio et al. [STT12] can be shown to imply that any polynomial p of weight W that 1/3-approximates the OR_d function at all Boolean inputs requires degree $\Omega(d/\log W)$.[3] The proof in [STT12] relies on a Markov-type inequality that bounds the derivative of a univariate polynomial in terms of its degree and the size of its coefficients. The proof of this Markov-type inequality is non-constructive and relies on complex analysis.

Here, we observe that our dual witness construction used to prove Corollary 5.2 also yields a general lower bound on the tradeoffs achievable between the weight and degree of the approximating polynomial p, even when we require p to be accurate only on inputs of Hamming weight at most $O(\log W)$ (see Theorem 5.8). The methods of Servedio et al. do not appear to yield any non-trivial lower bound on the degree in this setting. We also believe our proof technique is of interest in comparison to the methods of Servedio et al. as it is constructive (exhibiting an explicit dual witness for the lower bound) and avoids the use of complex analysis.

[3] More precisely, [STT12, Lemma 8] as stated shows that if the coefficients of a univariate polynomial P each have absolute value at most W, and $1/2 \leq \max_{x\in[0,1]} |P(x)| \leq R$, then $\max_{x\in[0,1]} |P'(x)| = O(\deg(P) \cdot R \cdot (\log W + \log\deg(P)))$, where $P'(x)$ denotes the derivative of P at x, and $\deg(P)$ denotes the degree of P. By inspection of the proof, it is easily seen that if the L_1-norm of the coefficients of P is bounded by W, then the following slightly stronger conclusion holds: $\max_{x\in[0,1]} |P'(x)| = O(\deg(P) \cdot R \cdot \log W)$. When combined with the symmetrization argument of [STT12], this stronger conclusion implies: any polynomial p of weight W that 1/3-approximates the OR_d function at all Boolean inputs requires degree $\Omega(d/\log W)$.

Theorem 5.8. *Any polynomial p of weight W that $1/6$-approximates the OR_d function at all Boolean inputs requires degree $d/2^{O(\sqrt{\log W})}$.*

Proof. As p is accurate on the entire Boolen hypercube, it is accurate on inputs of Hamming weight at most $\log W$. The theorem follows by setting $k = \log W$ in the statement of Corollary 5.2. □

6 Discussion

We gave a differentially private online algorithm for answering k-way marginal queries that runs in time $2^{o(d)}$ per query, and guarantees accurate answers for databases of size $\mathrm{poly}(d, k)$. More precisely, we showed that if there exists a polynomial of degree t and weight W approximating the OR function on Boolean inputs of Hamming weight up to k, then a variant of the private multiplicative weights algorithm can answer k-way marginal queries in time roughly $\binom{d}{t}$ per query and guarantee accurate answers on databases of size roughly $W\sqrt{d}$. To this end, we gave a new construction showing the existence of polynomial approximations to the OR function on inputs of low Hamming weight. Specifically, we showed that polynomials of weight $d^{.01}$ and degree $d^{1-\Omega(1/\sqrt{k})}$ exist.

Our algorithm for answering k-way marginals is essentially the same as in [HR10], but using a different set of base functions (specifically, the set of all low-degree parities), which leads to an efficiency gain. We note that our algorithm degrades smoothly to the private multiplicative weights algorithm as the degree of the promised polynomial approximation increases, and never gives a worse running time. This behavior suggests that our algorithm may lead to practical improvements even for relatively small values of d, for which the asymptotic analysis does not apply. In such cases one might use an alternative but similar analysis that shows the existence of a polynomial of degree $kd^{1-c/k}$ and weight d^c (for any $0 < c < k$) that exactly computes the d-variate OR function on inputs of Hamming weight at most k. Such a polynomial may be obtained as in our construction, by breaking the d-variate OR function into an OR of ORs, and using a degree k polynomial defined via polynomial interpolation, instead of a transformation of the Chebyshev polynomials, to approximate the outer OR on inputs of Hamming weight at most k.

Our lower bounds show that our polynomial approximation to the OR_d function on inputs of Hamming weight k is close to the best possible; in particular, we cannot hope to improve the running time on $\mathrm{poly}(d, k)$ size databases by giving approximating polynomials with better weight and degree bounds. We do not know if it is possible to do better by using different feature spaces (other than the set of all low-degree monomials) to uniformly approximate all disjunctions over d variables. We leave this question as an interesting direction for future work.

Acknowledgments. We thank Salil Vadhan for helpful discussions about this work.

References

[BCD$^+$07] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In Leonid Libkin, editor, *PODS*, pages 273–282. ACM, 2007.

[BDMN05] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In Chen Li, editor, *PODS*, pages 128–138. ACM, 2005.

[BLR08] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *STOC*, pages 609–618, 2008.

[BS92] Jehoshua Bruck and Roman Smolensky. Polynomial threshold functions, ac^0 functions, and spectral norms. *SIAM J. Comput.*, 21(1):33–42, 1992.

[BT13] Mark Bun and Justin Thaler. Dual lower bounds for approximate degree and markov-bernstein inequalities. *To appear in ICALP 2013. Pre-print available at: CoRR*, abs/1302.6191, 2013.

[CKKL12] Mahdi Cheraghchi, Adam Klivans, Pravesh Kothari, and Homin K. Lee. Submodular functions are noise stable. In Dana Randall, editor, *SODA*, pages 1586–1592. SIAM, 2012.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC '06*, pages 265–284, 2006.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210. ACM, 2003.

[DNR$^+$09] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil P. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC '09*, pages 381–390, 2009.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60. IEEE Computer Society, 2010.

[FK13] Vitaly Feldman and Pravesh Kothari. Learning coverage functions. *Manuscript*, 2013.

[GHRU11] Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. In *STOC '11*, pages 803–812, 2011.

[GRU12] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *TCC*, pages 339–356, 2012.

[HLM12] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. *NIPS '12*, 2012.

[HR10] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70. IEEE Computer Society, 2010.

[HRS12] Moritz Hardt, Guy N. Rothblum, and Rocco A. Servedio. Private data release via learning thresholds. In Dana Randall, editor, *SODA*, pages 168–187. SIAM, 2012.

[JT12] Prateek Jain and Abhradeep Thakurta. Mirror descent based database privacy. In Anupam Gupta, Klaus Jansen, José D. P. Rolim, and Rocco A. Servedio, editors, *APPROX-RANDOM*, volume 7408 of *Lecture Notes in Computer Science*, pages 579–590. Springer, 2012.

[Kla11] Hartmut Klauck. On arthur merlin games in communication complexity. In *Proceedings of the 26th Annual Conference on Computational Complexity (CCC)*, CCC, pages 189–199, 2011.

[KRSU10] Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam Smith, and Jonathan Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *STOC*, pages 775–784, 2010.

[KS04] Adam R. Klivans and Rocco A. Servedio. Toward attribute efficient learning of decision lists and parities. In John Shawe-Taylor and Yoram Singer, editors, *COLT*, volume 3120 of *Lecture Notes in Computer Science*, pages 224–238. Springer, 2004.

[LS13] Philip M. Long and Rocco A. Servedio. Low-weight halfspaces for sparse boolean vectors. In Robert D. Kleinberg, editor, *ITCS*, pages 21–36. ACM, 2013.

[Pat92] R. Paturi. On the degree of polynomials that approximate symmetric boolean functions. In *Proceedings of the twenty-fourth annual ACM Symposium on Theory of Computing*, pages 468–474, 1992.

[RR10] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *STOC '10*, pages 765–774, 2010.

[She09] Alexander A. Sherstov. The intersection of two halfspaces has high threshold degree. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '09, pages 343–362, 2009.

[She11] A. A. Sherstov. The pattern matrix method. *SIAM Journal on Computing*, 40(6):1969–2000, 2011.

[She12a] Alexander A. Sherstov. The multiparty communication complexity of set disjointness. In Howard J. Karloff and Toniann Pitassi, editors, *STOC*, pages 525–548. ACM, 2012.

[She12b] Alexander A. Sherstov. Strong direct product theorems for quantum communication and query complexity. *SIAM J. Comput.*, 41(5):1122–1165, 2012.

[STT12] Rocco Servedio, Li-Yang Tan, and Justin Thaler. Attribute-efficient learning and weight-degree tradeoffs for polynomial threshold functions. In *25th Annual Conference on Computational Learning Theory (COLT), JMLR Workshop and Conference Proceeding*, volume 23, pages 14.1–14.19, 2012.

[TUV12] Justin Thaler, Jonathan Ullman, and Salil P. Vadhan. Faster algorithms for privately releasing marginals. In *ICALP (1)*, pages 810–821, 2012.

[Ull12] Jonathan Ullman. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. *CoRR*, abs/1207.6945, 2012.

[UV11] Jonathan Ullman and Salil P. Vadhan. PCPs and the hardness of generating private synthetic data. In *TCC '11*, pages 400–416, 2011.

[Vad] Salil Vadhan. Personal communication.

[Š08] Robert Špalek. A dual polynomial for OR. *CoRR*, abs/0803.4516, 2008.